O varianta extinsa e disponibila in cartea
'Proiectarea siturilor Web', Polirom, 2002:
www.infoiasi.ro/~design/

XHTML: inceputul sfirsitului?!

Sabin-Corneliu Buraga

Articol aparut in PC Report vol.9, 04 (92), aprilie 2000

La data de 26 ianuarie 2000, Consortiul Web a facut publica o noua specificatie-recomandare intitulata **XHTML**, reprezentind o familie, bazata pe XML, de tipuri de documente si module care extinde arhicunoscutul standard HTML. XHTML 1.0 este primul "descendent" al acestei familii, fiind in fapt reformularea tipurilor de documente HTML 4 in termenii meta-limbajului XML. Dezvoltatorii de pagini si de aplicatii Web vor migra astfel de la HTML 4, bazat pe relativ vechiul si complexul SGML (Standard Generalized Markup Language), la XML (eXtensible Markup Language) - extensibil, tinar si mai usor de utilizat, cu un viitor intrevazut a fi stralucit.

Motive de a rescrie HTML in XML

Motivele rescrierii limbajului HTML (a carui ultima specificatie HTML 4.01 a aparut la sfirsitul lui 1999) in termenii XML sint date de mai multi factori:

1. **Extensibilitatea**

   XML este un limbaj de marcare cu adevarat extensibil, oferind puterea si flexibilitatea de specificare a SGML, prin mijloace facile. Cerinta HTML este de a se adapta la piete electronice, specializate si in plina dezvoltare, foarte dinamice. Aceasta conduce la probleme de compatibilitate intre documente de pe diverse platforme hardware si software, solutionare intrevazuta in XML.

2. **Modularizarea**

   Modularizarea presupune o metoda de specificare a unor multimi bine-definite de seturi de tag-uri HTML spre a fi utilizate de designerii paginilor Web. De exemplu, un modul "tabele" poate contine elementele si atributele necesare pentru realizarea tabelelor, iar un model "liste" poate ingloba elementele si atributele pentru liste. In functie de aplicatiile Web vizate, vor putea fi folosite diverse module, prin restrictii sau extensii ale XHTML.

   Modularizarea HTML-ului isi poate gasi aplicatii si in portarea navigarii pe Web la nivelul dispozitivelor mobile (calculatoare portabile, telefoane celulare) sau aparatelor TV (televiziune digitala, TV Web). Fiecare categorie comporta diferite cerinte si restrictii (specializari).

   Modularizarea poate creste si productivitatea si standardizarea in realizarea documentelor Web.

3. **Profile ale documentelor**

Profilul documentelor specifica sintaxa si semantica acestora. Profilul va include specificatii ca: multimea formatelor suportate (e.g. formatele tipurilor de media (imagini, sunete,...) ce pot fi utilizate), nivelul limbajelor script si suportul pentru foi de stil etc.

Profilul documentelor este in strinsa legatura cu RDF (Resource Description Framework) pe care l-am prezentat intr-un articol anterior, constind din asertiuni scrise in RDF definind suportul minim al agentilor utilizator si oferind o baza pentru garantarea interoperabilitatii. Schemele RDF vor formaliza profilele documentelor.

Aceste asertiuni se asteapta sa aiba urmatoarele efecte:

- garantia longevitatii profilului unui document.
- sintaxa documentului data ca legatura specificata de un URI. O schema defineste aceasta sintaxa drept compunere de multimi de tag-uri (module).
- reprezentarea restrictiilor semantice ale documentelor va fi interpretabila de catre masina. Aceste restrictii sint folosite la validarea documentelor si vor fi introduse in urmatoarele specificatii XML (scheme si date XML).
- asertiunile vor putea acoperi diverse detalii ale formatelor de date si vor descrie tipurile de dispozitive potrivite cu profilul unui document.

4. **Profile ale dispozitivelor**

O directie separata este de a folosi RDF pentru definirea profilelor dispozitivelor, specificind capabilitatile navigatoarelor si preferintele utilizatorilor.

5. **Transformarea marcajelor**

Profilele documentelor si ale dispozitivelor vor aduce mari simplificari in marcarea informatiilor. Atunci cind un set de facilitati HTML suportate de o clasa de dispozitive poate fi anticipat cu precizie, marcarea specifica pentru acea clasa hardware se realizeaza intr-un mod facil si automatizat.

Specificatia XHTML 1.0

XHTML (sau pina anul trecut **Voyager**) este numele de cod al HTML reformulat ca aplicatie a XML-ului, definind profilele documentelor ca spatii de nume, avind fiecare propria sa adresa (data de un URI).

Documentele XHTML pot fi etichetate atit "text/html" cit si "text/xml", elementul radacina fiind <html>. Spatiile de nume XHTML sint definite la adresa http://www.w3.org/1999/xhtml si vor putea fi incluse in documentul XHTML prin atributul xmlns.

Declarea tipului de documente XHTML se va face prin intermediul constructiei DOCTYPE , ca in SGML, existind trei tipuri de definitii de documente conform specificatiei

O varianta extinsa e disponibila in cartea
'Proiectarea siturilor Web', Polirom, 2002:
www.infoiasi.ro/~design/

HTML 4: tipul *strict*, *tranzitional* si *pentru cadre (frames)*, ca in exemplul urmator:

```
<!DOCTYPE html
        PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "DTD/xhtml1-strict.dtd">
```

Modelul procesarii

Documentele XHTML sint procesate in urmatorii pasi:

1. decodificarea datelor preluate (via protocolul HTTP) de pe server;
2. descompunerea in atomi lexicali XML 1.0;
3. analiza sintactica XML, rezultind un arbore sintactic ulterior accesat si manipulat prin *DOM (Document Object Model)*;
4. validare optionala;
5. formatare, folosind foi de stiluri si alte semantici specificate (pentru formulare si applet-uri), pentru a produce o ierarhie a obiectelor formatate;
6. reprezentarea (pe un anumit suport) a obiectelor formatate.

Navigatoarele executa acesti pasi intr-o maniera concurenta.

Documentele HTML pot fi convertite in documente XHTML de **Tidy**, aplicatie dezvoltata si pusa gratuit la dispozitie de Consortiul WWW.

Module XHTML

XHTML ofera mai mult decit o reformulare a HTML-ului in XML, modularizind HTML printr-o colectie de multimi de tag-uri, blocuri de constructie a produselor Web. In prezent, modularizarea limbajului HTML este in derulare, in urmatoarele luni dindu-se publicitatii varianta finala.

Modulele propuse sint cele din tabelul alaturat:

Sfaturi pentru dezvoltatori

XHTML fiind bazat pe XML este *case-sensitive*, tag-urile trebuie sa fie scrise cu caractere mici. Daca agentul-utilizator (browserul) nu recunoaste un element, va prelucra continutul lui aflat intre tag-urile de inceput si de sfirsit. Un atribut nerecunoscut va fi, de asemeni, ignorat, iar o valoare necunoscuta a unui atribut va fi inlocuita cu valoarea implicita a acestuia.

Atributul de tip ID (identificator in XML) va fi considerat ca

Modulele XHTML

- **modulul de baza**
  specifica tipurile de date si modele de baza XHTML, impreuna cu setul minimal de elemente pentru scrierea unui document HTML: html, head, title, base, meta, link, body, h1-h6, p, br, a, span, div.
- **modulul tranzitional**
  specifica acele elemente din HTML 4.0 - profilul tranzitional, dar excluse din profilul strict al HTML: basefont, font, center, s, u, continind si definitiile unor atribute ca border, align, noshade.
- **modulul de stiluri**

fragment de document. Elementele a (ancora), applet (applet Java), form (formular), iframe (cadru intern), frame (cadru), img (imagine), map (harta senzitiva) in HTML pot avea (obligatoriu sau nu) atributul name. In XHTML 1.0 este definit atributul id de tip ID pentru acest scop si name va fi inlocuit in viitor cu atributul id. Pentru compatibilitate cu HTML 4.0 se vor folosi ambele atribute:

```
<a id="top" name="top"></a>
...
<h6><a href="#top">La
  inceputul paginii</a></h6>
```

De remarcat faptul ca toate valorile atributelor (chiar si cele numerice) vor trebuie incluse intre ghilimele. Astfel, constructii precum <table width=600 align=right> sint eronate.

Anumite atribute in HTML puteau fi scrise fara a le asocia valori (asa-numitele atribute booleene) i.e. noshade, readonly, noresize, compact ori checked . In XHTML vor trebui scrise succedate de valorile lor, ca in exemplul urmator:

```
<hr
  size="2"
  align="right"
  noshade="noshade" />
```

Alta diferenta fata de HTML 4.0 este aceea ca elementele nevide trebuie sa aiba tag-uri de sfirsit. In SGML anumite elemente pot avea tag-uri de sfirsit optionale. In XML, pentru a elimina ambiguitati de analiza, elementele trebuie sa aiba tag-uri de sfirsit obligatorii. Astfel, paragrafele marcate prin intermediul lui vor avea si tag-ul de sfirsit . Elementele declarate cu continut vid, precum <hr> sau <br>, fie vor fi urmate de tag-ul de sfirsit, fie vor fi scrise <hr /> ori, respectiv, <br />. Insa este interzis sa avem din moment ce acest element nu este declarat ca vid (prin EMPTY).

ofera elementul style, atributul style si defineste utilizarea elementului link.

- **modulul script**
  specifica elementele script si noscript.
- **modulul de fonturi**
  specifica elementele relative la fonturi existente in HTML 4.0 - varianta stricta: tt, b, i, big, small.
- **modulul de frazare**
  defineste elementele care ofera informatii despre anumite adnotari ale autorului, in cadrul frazei: abbr, acronym, address, blockquote, q, cite, code, dfn, kbd, samp, var.
- **modulul de inflexiune**
  specifica acele elemente frazale care nu ofera informatii referitoare la un domeniu ci doar o indicatie despre intentia autorului: em, pre, strong, sub, sup, hr.
- **modulul de editare**
  ofera elemente referitoare la editarea documentelor, ca del si ins.
- **modulul liste**
  specifica elementele dl, dt, dd, ul, ol, li.
- **modulul de formulare**
  specifica elementele referitoare la scrierea formularelor: form, input, textarea, select, optgroup, option, label, button, fieldset, legend, isindex.
- **modulul de tabele**
  contine elementele table, caption, col, colgroup, thead, tbody, tr, th, td.
- **modulul harti de imagini**
  contine elementele map si area.
- **modulul applet-uri**
  contine applet si param pentru suportul applet-urilor Java.
- **modulul obiecte**
  specifica elemente ca object si param.
- **modulul cadre**
  specifica elementele referitoare la utilizarea cadrelor: frameset, frame, iframe, noframes.

Desi navigatoarele treceau cu vederea proasta imbricare a tag-urilor, specificatia XHTML este mai riguroasa. Elementele trebuie sa aiba deci tag-uri de inceput si de sfirsit scrise corect. Constructii ca <h4>XHTML este exact.</h4> sint prohibite.

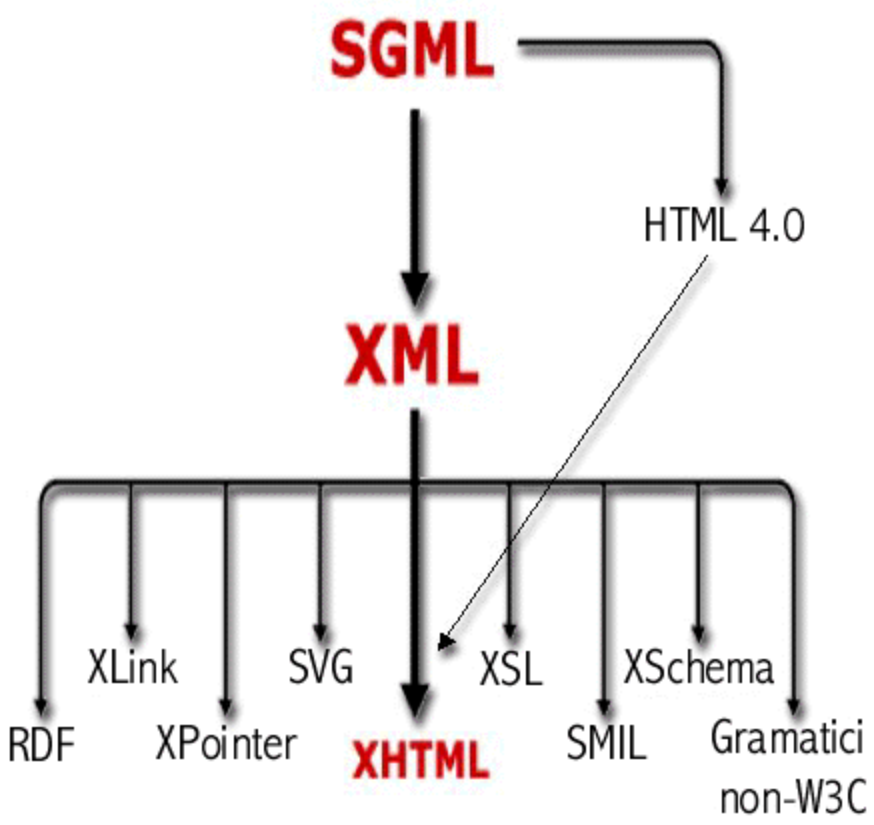
Desi in XML nu pot fi specificate incluziuni si excluziuni in cadrul definitiilor formale de elemente, asa cum se intimpla in SGML, se impun urmatoarele restrictii:

- Elementele a, form, label nu pot contine alte elemente a, form, label, respectiv.
- Elementul pre nu poate contine img, object, big, small, sub si sup.
- Elementul button nu poate sa contina elementele input, select, textarea, label, button, form, fieldset, iframe sau isindex.
- Elementul title trebuie sa apara imediat dupa html.
- Elementul isindex poate apare cel mult o data in cadrul antetului documentului si este declarat demodat, fiind inlocuit in prezent de input.

Designerii trebuie sa utilizeze foi de stiluri sau programe script externe daca acestea cuprind simbolurile < sau & sau]]> sau --, din cauza modelului nou de procesare XML a documentelor XHTML. Pentru script-urile CGI (Common Gateway Interface) parametrii transmisi vor trebui sa aiba caracterul ampersand inlocuit de entitatea &. Astfel, linia http://www.infoiasi.ro/cgi-bin/search.pl?name=Sabin&value=Circles va trebui inlocuita de http://www.infoiasi.ro/cgi-bin/search.pl?name=Sabin&value=Circles.

Utilizatorii modelului obiectual de documente (DOM) trebuie sa tina cont de urmatoarele: recomandarea pentru DOM nivelul 1 defineste interfete atit pentru XML cit si pentru HTML 4, DOM pentru HTML specificind ca numele de elemente si de atribute vor fi returnate cu caractere mari, iar pentru XML vor fi returnate asa cum au fost specificate de creatorul documentului. Aceste diferente vor putea fi rezolvate in functie de tipul MIME returnat: aplicatiile care manipuleaza documentele XHTML prin intermediul tipului text/html vor utiliza DOM pentru HTML, iar cele folosind tipul text/xml sau application/xml vor utiliza DOM pentru XML.

Un exemplu complet de documente conforme cu standardul XHTML 1.0 ar trebui la momentul aparitiei acestui text sa fie disponibil la adresa http://www.infoiasi.ro/fcs/index.html.



Arborele genealogic al XHTML

Concluzii

Meta-limbajul SGML a fost standardizat in 1986 si probabil cea mai insemnata aplicatie a lui a reprezentat-o HTML-ul, propus la sfirsitul deceniului 9 ca limbaj de marcare a informatiilor hipertext, ulterior adaugindu-i-se si capabilitati multimedia. Din cauza complexitatii majore, SGML nu a avut un succes notabil printre utilizatorii frecventi, fiind folosit in special de specialistii in lingvistica computationala sau de proiectantii de editoare/procesoare de documente sofisticate.

Situatia s-a schimbat odata cu aparitia XML-ului, standardizat in parte doar acum 2-3 ani de Consortiul Web si sprijinit de multe companii si organizatii, in prezent intr-o dezvoltare fara precedent. XML a dovedit punctele slabe ale limbajului HTML: lipsa de adaptabilitate si flexibilitate si de suport pentru programare si modularizare, necesitatea rescrierii conform filosofiei XML fiind vitala pentru proiectantii de aplicatii Web. XHTML 1.0 este doar inceputul unui proces complex de schimbari, modalitatile alternative de accesare a Internetului fiind deja intrevazute de futurologii spatiului virtual. Familia XHTML a fost gindita sa sprijine interoperabilitatea navigatoarelor Web multi-platforma care vor trebui sa se adapteze unei piete mondiale electronice in plin dinamism. Cert este ca epoca HTML-ului clasic este la final.

Sabin-Corneliu Buraga este doctorand in Computer Science la Universitatea "A.I.Cuza" din Iasi, putind fi contactat la adresa busaco@infoiasi.ro.

O varianta extinsa e disponibila in cartea
'Proiectarea siturilor Web', Polirom, 2002:
www.infoiasi.ro/~design/